Spatial Demography ***2014*** 2(1): 72-78
http://spatialdemography.org

SPATIAL DEMOGRAPHY
ISSN 2164-7070 (online)

COLUMN

Measuring Residential Segregation using R

So Long to Factfinder

Corey Sparks
Editor, Software & Code

In this installment of the "Software and Code" column, I use the open source statistical programming environment R to calculate commonly used indices of residential segregation. Such measures are commonly used in spatial demography as both outcome and as predictor. The social demography literature has a rich history of the dimensions of residential segregation (Massey & Denton, 1988; Reardon & Firebaugh, 2002), and I will not attempt to review this literature in the current setting. Instead, I will focus on using the programming structures and user-written libraries in R to show users how quickly and easily they can calculate some of the most common indices of residential segregation. This being said, some common indices are easily accessible through the University of Michigan's Population Center (http://enceladus.isr.umich.edu/race/racestart.asp), but often researchers need more flexibility in calculating their own group-specific measures of segregation, which are not commonly available.

In previous columns, I have relied on the US Census Bureau's American Factfinder website for all data acquisition. In this column, I turn to using a contributed package for R named Uscensus-suite of packages, including the updated UScensus2010 packages (Almquist, 2010). These packages are excellent because they allow access to the 2000 and 2010 Census Summary File 1 data at geographies down to the block level for the 2010 census and the block group level for the 1990 and 2000 Census. This removes the need for using the often awkward Factfinder website, because these packages allow the user to download Summary File tables directly into an R data frame.

In the present example, I will calculate the indices of black-white dissimilarity, D, and interaction ${}_xP_y$, the black isolation index, ${}_xP_x$, and the multi-group Theil H index for blacks, whites and those of all other races (3 groups) at the county level for the state of Texas using the 2010 Summary File 1 data. To to these calculations, I will aggregate up from the Census tract level to the county, although one could easily aggregate up from the block or block group level using this package. Then I will show how to merge these to county-level indices to county shapefiles, also contained in the UScensus2010 package.

To begin, I must first install the UScensus2010 package, then install the tract data. It should be noted that if users are working in a Windows environment, they should install the Rtools package (http://cran.r-project.org/bin/windows/Rtools/) because it contains compilers necessary to build the packages I are going to use. It should also be noted that the tract, block group and block data are very large, and can take some time to download, I would not recommend one try this over a wireless connection. The block data are around 4GB in size, while the tract data are just a few hundred MB in size.

install.packages("UScensus2010",

```
 dependencies=T)
library(UScensus2010)
install.tract("osx")
install.county("osx")
library(UScensus2010tract)
library(UScensus2010county)
```

The *install.tract("osx")* command has R download the tract data, since I am working on a Mac platform, I use "osx" as my install option. I would generally recommend users of Windows to use the *"linux"* option (I know, it says linux, but trust me), because these data are compiled from source code and are not natively available for Windows users, sorry.

First I create a set of three race variables, total, white and black for each census tract in the state of Texas from the 2010 SF1 data. The table I need is the P003 table (See a good list of tables here: http://www.socialexplorer.com/data/C2010/metadata/?ds=SF1), which is population by race. The first step is to download the total population. The *demographics()* function is used to extract the total population for tracts from the state of Texas.

```
mydem<-demographics(dem="P0030001",
state="48", statefips=TRUE, level="tract")
```

Then, I assemble a data frame with sensical names and create a county FIPS code for each tract. But first, I assemble the "other" race group, which in the SF1 is composed of 5 tables P0030004 to P0030008.

```
other<-data.frame(
 cofips=substr(rownames(mydem), 1,5),
 fips=rownames(mydem),

oth1=unlist(demographics(dem="P0030004",st
ate="48",statefips=T, level="tract")),

oth2=unlist(demographics(dem="P0030005",st
ate="48",statefips=T, level="tract")),

oth3=unlist(demographics(dem="P0030006",st
ate="48",statefips=T, level="tract")),

oth4=unlist(demographics(dem="P0030007",st
ate="48",statefips=T, level="tract")),

oth5=unlist(demographics(dem="P0030008",st
ate="48",statefips=T, level="tract"))
)
```

Next, I build the tract population data for all races.

```
trdat<-
data.frame(cofips=substr(rownames(mydem),
1,5),
        fips=rownames(mydem),

total=unlist(demographics(dem="P0030001",st
ate="48",statefips=T, level="tract")),

white=unlist(demographics(dem="P0030002",s
tate="48",statefips=T, level="tract")),

black=unlist(demographics(dem="P0030003",s
tate="48",statefips=T, level="tract")),
        other=apply(other[,3:7],1,sum))
```

I sort the data by county and tract
```
trdat<-trdat[order(trdat$cofips, trdat$fips),]
```

Then I assign some nice names

```
names(trdat)<-c("cofips", "fips", "total", "white",
"black", "other")
```

And I look at the first few cases. Note I have 5,265 tracts with 6 variables at this point.

```
head(trdat)
```

cofips	fips	total	white	black	other
48001	48001950100	4685	4012	452	221
48001	48001950401	5422	1825	2266	1331
48001	48001950402	7535	2591	3248	1696
48001	48001950500	4377	2737	800	840
48001	48001950600	6405	3831	1674	900
48001	48001950700	2640	1051	1164	425

Here, cofips is the county FIPS code, fips is the tract FIPS code, total is the total population of the tract, white is the white population of the tract, black is the black population of the tract and other is the population of all other races in the tract.

Segregation indices are generally a combination of larger-geography data (say, at the county-level) and lower level geography data (say, the tract level), so I need the county-level totals for the total population and each race group. This is straight forward to do using the *tapply()* function, which applies a function a function to a variable across a group variable. Here I will sum the populations at the tract level up to the county level.

```
co_total<-tapply(trdat$total, trdat$cofips, sum)
co_total<-
data.frame(cofips=names(unlist(co_total)),
pop=unlist(co_total))
co_wht<-tapply(trdat$white, trdat$cofips, sum)
co_wht<-
data.frame(cofips=names(unlist(co_wht)),
pop=unlist(co_wht))
co_blk<-tapply(trdat$black, trdat$cofips, sum)
co_blk<-
data.frame(cofips=names(unlist(co_blk)),
pop=unlist(co_blk))
co_oth<-tapply(trdat$other, trdat$cofips, sum)
co_oth<-
data.frame(cofips=names(unlist(co_oth)),
pop=unlist(co_oth))
```

For the multi-group measure of segregation, I also need population proportions, it is easier to do now versus later.

```
c_pwhite<-co_wht$pop/co_total$pop
c_pblack<-co_blk$pop/co_total$pop
c_pother<-co_oth$pop/co_total$pop
```

Next, I assemble them into a county-level data frame with easy names

```
county_dat<-
data.frame(cofips=co_total$cofips,
co_total=co_total$pop,
co_wht_total=co_wht$pop,
co_blk_total=co_blk$pop,
co_oth_total=co_oth$pop, c_pwhite=c_pwhite,
c_pblack=c_pblack, c_pother=c_pother)
```

One of our indices, Theile's H, requires a county-level measure of diversity, called entropy. Now I make the county-level Entropy measure to be used later, it's easier to do it before merging the county data back to the tract data.

```
county_dat$c_ent<-
county_dat$c_pwhite*(log(1/county_dat$c_pwhite))
+county_dat$c_pblack*(log(1/county_dat$c_pblack))
+county_dat$c_pother*(log(1/county_dat$c_pother))
county_dat$c_ent<-
ifelse(is.na(county_dat$c_ent)==T,
0,county_dat$c_ent)
```

Now, I merge the county data back to the tract data by the county FIPS code

```
merged<-merge(x=trdat,y=county_dat,
by="cofips", all.x=T )
```

And I always have to have a look and make sure it looks ok

```
head(merged)
```

Which shows us the tract data, but with the county totals merged to it. In general, this is the form of a data set that is needed to make segregation indices.

cofips	fips	total	white	black	other	co_total	co_wht_total	co_blk_total	co_oth_total	c_pwhite	c_pblack	c_pother	c_ent
48001	48001950100	4685	4012	452	221	58458	38632	12310	7516	0.661	0.211	0.129	0.866
48001	48001950401	5422	1825	2266	1331	58458	38632	12310	7516	0.661	0.211	0.129	0.866
48001	48001950402	7535	2591	3248	1696	58458	38632	12310	7516	0.661	0.211	0.129	0.866
48001	48001950500	4377	2737	800	840	58458	38632	12310	7516	0.661	0.211	0.129	0.866
48001	48001950600	6405	3831	1674	900	58458	38632	12310	7516	0.661	0.211	0.129	0.866
48001	48001950700	2640	1051	1164	425	58458	38632	12310	7516	0.661	0.211	0.129	0.866

2 group segregation measures

Now I begin the segregation calculations
First, I calculate the tract-specific contribution to the county dissimilarity index

```
merged$d.wb<-.5*(abs(merged$white/merged$
co_wht_total -
merged$black/merged$co_blk_total))
```

The county-level dissimilarity index is just the sum of the tract values within a county, this is easily done with the *tapply()* function, which in this case will sum the tract-specific contributions to the index within each county.

```
dissim.wb.tr<-tapply(merged$d.wb,
merged$cofips, sum, na.rm=T)
head(dissim.wb.tr)
```

48001	48003	48005	48007	48009	48011
0.4377915	0.2227811	0.46463	0.2247782	0.3273754	0

Next is the black-white interaction index. In this calculation, the first population is minority population and the second is non-minority population.

```
merged$int.wb<-
(merged$black/merged$co_blk_total *
merged$white/merged$total)
```

Again, I sum these within each county

```
int.wb.tr<-tapply(merged$int.wb,
merged$cofips, sum, na.rm=T)
head(int.wb.tr)
```

48001	48003	48005	48007	48009	48011
0.5198444	0.7840112	0.54278	0.85285	0.9498711	0.9331931

Next is the isolation index for blacks.

```
merged$iso.b<-
(merged$black/merged$co_blk_total *
merged$black/merged$total)
```

Again, I sum these within each county

```
isol.b.tr<-tapply(merged$iso.b, merged$cofips,
sum, na.rm=T)
head(isol.b.tr)
```

48001	48003	48005	48007	48009	48011
0.308	0.018	0.290	0.018	0.007	0.006

Multi-group segregation

Here, I calculate the Theile Entropy index. Just like for the counties, I also need population proportions at the tract level.

```
merged$tr_pwhite<-
merged$white/merged$total
merged$tr_pblk<-merged$black/merged$total
merged$tr_pother<-
merged$other/merged$total
```

Here I calculate the tract-level entropy measure

```
merged$tr_ent<-
merged$tr_pwhite*(log(1/merged$tr_pwhite))
+merged$tr_pblk*(log(1/merged$tr_pblk))
+merged$tr_pother*(log(1/merged$tr_pother))
merged$tr_ent<-
ifelse(is.na(merged$tr_ent)==T,
0,merged$tr_ent)
```

Now I calculate each tract's contribution to the H index

```
merged$Hcalc<-(merged$total*(merged$c_ent-
merged$tr_ent))/
(merged$co_total*merged$c_ent)
merged$Hcalc<-
ifelse(is.na(merged$Hcalc)==T, 0,
merged$Hcalc)
```

Then, I sum across tracts within counties to get the H index for each county

```
hindex<-unlist(tapply(merged$Hcalc,
merged$cofips, sum))
```

Finally, I assemble all of the indices into a dataframe with nice names

```
county_dat<-
data.frame(cofips=names(unlist(dissim.wb.tr)),
```

dissim_wb=unlist(dissim.wb.tr), isolation_b=isol.b.tr, interaction_bw=int.wb.tr, TheileH=hindex)

And I have a look at the indices

head(county_dat)

cofips	dissim_wb	isolation_b	interaction_bw	TheileH
48001	0.438	0.308	0.520	0.120
48003	0.223	0.018	0.784	0.013
48005	0.465	0.290	0.543	0.137
48007	0.225	0.018	0.858	0.014
48009	0.327	0.007	0.950	0.032
48011	0.000	0.006	0.933	0.000

Next, I get a numeric summary of them. I primarily do this to make sure they are all bound on 0,1, which they should be.
summary(county_dat[, 2:5])

dissim_wb	isolation_b	interaction_bw	TheileH
Min. :0.0000	Min. :0.00000	Min. :0.0000	Min. :0.000000
1st Qu.:0.1054	1st Qu.:0.01492	1st Qu.:0.6910	1st Qu.:0.005786
Median :0.2642	Median :0.05619	Median :0.7837	Median :0.031409
Mean :0.2586	Mean :0.09986	Mean :0.7569	Mean :0.046053
3rd Qu.:0.4034	3rd Qu.:0.14921	3rd Qu.:0.8671	3rd Qu.:0.069289
Max. :0.7229	Max. :0.55264	Max. :0.9499	Max. :0.335423

And I plot histograms of all the indices.

par(mfrow=c(2,2))
hist(county_dat$dissim_wb, main="White-Black Dissimilarity")
hist(county_dat$isolation_b, main="Black Isolation")
hist(county_dat$interaction_bw, main="White-Black Interaction")
hist(county_dat$TheileH, main="White-Black-Other Theile H Index")

Now I need to do some visualization of the indices, so I load a US County shapefile, which is contained in the UScensus2010county library (no need to go get them from Census!), the Texas county SF1 data is in texas.county10, it also has the county polygons.
data("texas.county10")

I only want the identifiers, so I get rid of the SF1 data:

tx_sub<-texas.county10[, c(1:6)]

I put the attribute table into a new object

tx_attr<-tx_sub@data

Then I merge the county attribute table to the segregation indices by county FIPS code. I am careful here not to re-sort the data, they need to be in the same order as the shapefile, and I want to keep all counties, even if they're missing the indices (which none are in this case)

merge2<-merge(x=tx_attr, y=county_dat, by.x="fips", by.y="cofips", all.x=T, sort=F)

Next, I re-attach the attribute table to the shapefile

tx_sub@data<-merge2

Next I map the indices, using quantile breaks, and using a nice color ramp from ColorBrewer

library(sp)
library(RColorBrewer)
brks<-quantile(tx_sub$TheileH, probs=c(0,.25, .5, .75, 1))
cols<-brewer.pal(n=5, name="Reds")

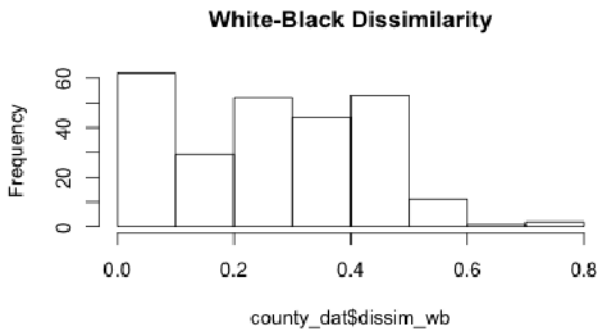
I want to save this file to an image, so I name where I want that to be, and the resolution I want:

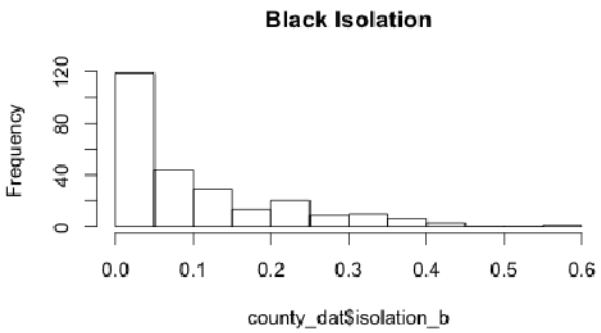
png(filename="~/Dropbox/spatialDemography /column3/TexasSegmap.png", res=150, width=900, height=900)
spplot(obj=tx_sub, zcol="TheileH",at=brks, col.regions=cols , main="Three-Race Theile H Index, 2010")

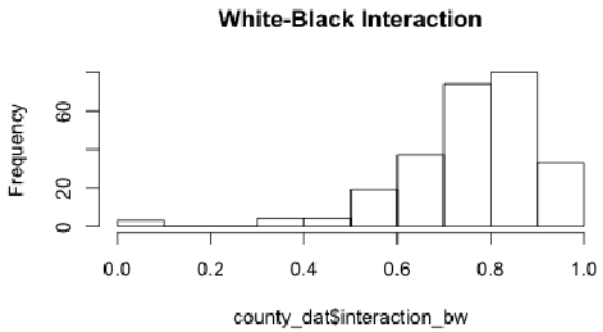
dev.off()

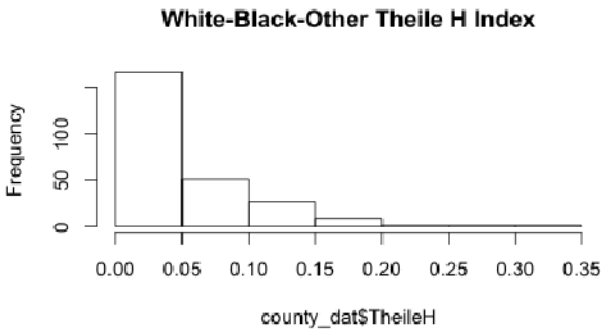
Finally, I calculate the spatial autocorrelation in one of the indices (See Column 1 for more of this type of thing)

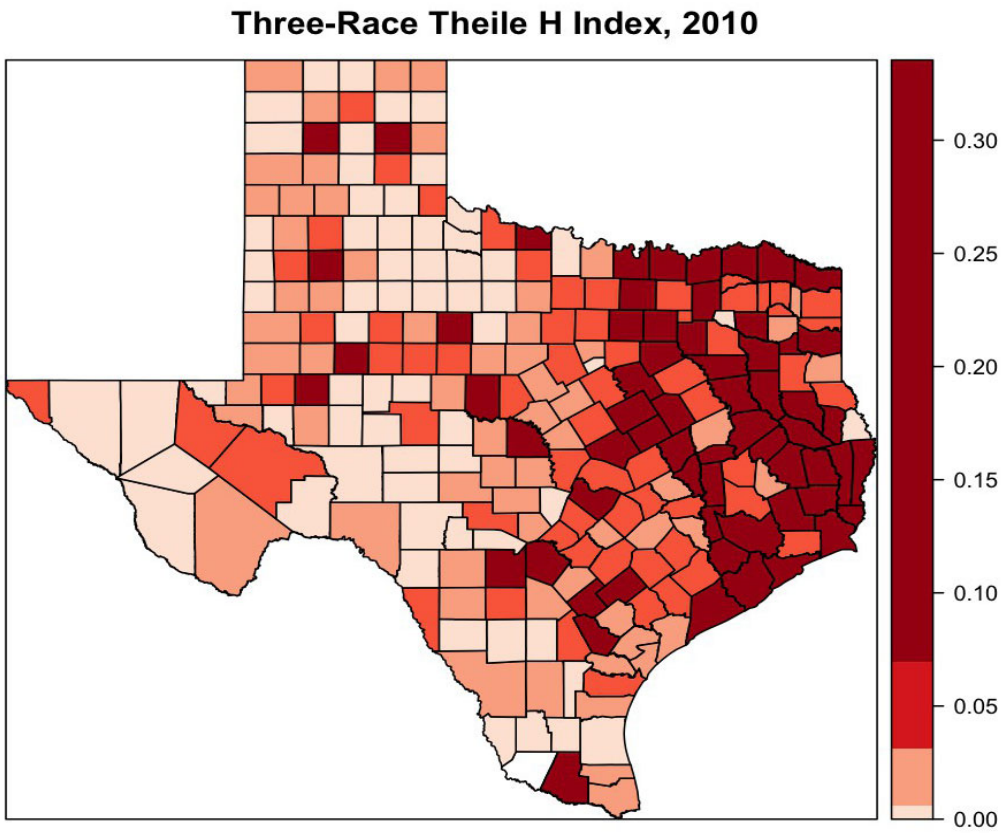
library(spdep)
nbs<-poly2nb(tx_sub, queen=T)


White-Black Dissimilarity
Frequency
0
20
40
60
0.0
0.2
0.4
0.6
0.8
county_dat$dissim_wb


Black Isolation
Frequency
0
40
80
120
0.0
0.1
0.2
0.3
0.4
0.5
0.6
county_dat$isolation_b


White-Black Interaction
Frequency
0
20
60
0.0
0.2
0.4
0.6
0.8
1.0
county_dat$interaction_bw


White-Black-Other Theile H Index
Frequency
0
50
100
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
county_dat$TheileH


Three-Race Theile H Index, 2010
0.30
0.25
0.20
0.15
0.10
0.05
0.00

```
wts<-nb2listw(nbs, style="W")
moran.mc(tx_sub$TheileH, listw=wts,
nsim=999)

        Monte-Carlo simulation of Moran's I

data:  tx_sub$TheileH
weights: wts
number of simulations + 1: 1000

statistic = 0.3527, observed rank = 1000, p-value
= 0.001
alternative hypothesis: greater

geary.mc(x=as.numeric(tx_sub$dissim_wb),
listw=wts, nsim=999)

        Monte-Carlo simulation of Geary's C

data:  as.numeric(tx_sub$dissim_wb)
weights: wts
number of simulations + 1: 1000

statistic = 0.7146, observed rank = 1, p-value =
0.001
alternative hypothesis: greater
```

Summary

In the above example, I have shown how to use packages within R to calculate several commonly used measures of residential segregation, map them and assess spatial autocorrelation within them without having to interact with the Census American Factfinder website, or having to download any external data. Again, this shows the flexibility of the R programming environment for uses in spatial demographic analysis. While the examples above use traditional black-white segregation, any groups may be compared using these methods, and all within the R environment. In future columns, I will illustrate the use of R for merging these data to individual level survey data for applications in multi-level modeling, without having to resort to closed-source proprietary software, such as HLM, MLWin, or SAS.

References

Almquist, Z. W. (2010). US Census Spatial and Demographic Data in R: The UScensus2000 Suite of Packages. *Journal of Statistical Software*, *37*(6), 1–31. Retrieved from http://www.jstatsoft.org/v37/i06

Massey, D. S., & Denton, N. A. (1988). The Dimensions and Residential Segregation. *Social Forces*, *67*, 281–315. doi:10.1093/sf/67.2.281

Reardon, S. F., & Firebaugh, G. (2002). Measures of Multigroup Segregation. *Sociological Methodology*, *32*, 33–67. doi:10.1111/1467-9531.00110